LBNL-48393
DR-432

To be presented at the Seventh International IBPSA Conference Building Simulation 2001, August 13-15, 2001 in Rio de Janeiro, Brazil and to be published in the Proceedings.

INCORPORATING LCA TOOLS IN INTEGRATED SIMULATION ENVIRONMENTS

Vineeta Pal, Konstantinos Papamichael, Norman Bourassa, and John. J. Loffeld

Building Technologies Department
Environmental Energy Technologies Division
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
1 Cyclotron Road
Berkeley CA 94720, U.S.A.

February 2001

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology, State and Community Programs, Office of Building Research and Standards of the U.S. Department of Energy.

INCORPORATING LCA TOOLS IN INTEGRATED SIMULATION ENVIRONMENTS

Vineeta Pal, Konstantinos Papamichael, Norman Bourassa, and John. J. Loffeld

Building Technologies Department, Environmental Energy Technologies Division
Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley CA 94720, U.S.A.

ABSTRACT

In this paper we address the issue of building data schema evolution in integrated simulation environments, as seen from the perspective of incorporating LCA tools within these environments. First we describe the key features of an integrated simulation environment designed for expandability, focusing on a) the mechanism for the expansion of the integrated environment, and b) its overall system architecture that allows processes and data to be added to the system without modifications or restructuring of existing code.

We then focus on how the data schema allows the inclusion and maintenance of specialized construction objects bearing LCA data. Finally, we discuss various integration issues that arise from modeling capabilities and idiosyncrasies of individual simulation and analysis tools.

INTRODUCTION AND BACKGROUND

During the past three decades, several building simulation tools have been developed to address aspects of building performance, including occupant comfort, energy use, costs, and environmental impact (Ward and Shakespeare, 1998; Trusty and Meil, 1997; Birdsall et al., 1990; Klein et al., 1976). Many of these tools were developed primarily for use by researchers, so user friendliness was not a high priority in their design. Attention was focused on producing detailed modeling capabilities and accurate results. However, as computer use has become common and professionals such as architects and engineers can profit from access to sophisticated building simulation models, more attention has been paid to the "front ends" of these tools to encourage the widespread design of energy-efficient, comfortable buildings. During the past 10 years, the building simulation industry has also attempted to integrate various building tools into one software environment that can model not just individual aspects of building design but also their inter-dependencies (Papamichael et al., 1997; Jokela et al., 1997; Mahdavi et al., 1996; Pohl et al., 1992). This task has proven challenging, mainly because the semantic representation of a building can vary drastically depending upon the aspect of the building being modeled. For example, walls could be represented as "thermal barriers" with "u-values" and "areas" for thermal computations or as "polygons" with "reflectance" values and "textures" for lighting computations.

Recent developments in information and tool sharing over the Internet make the attempts to integrate building software even more important. Manufacturers' data on building components may be available in a continuously updated form on the Internet but it needs to be in a format that is compatible with the various tools that use it. The International Alliance for Interoperability (IAI) is developing the Industry Foundation Classes (IFCs) (http://iaiweb.lbl.gov/) partly in response to these issues, as well as to allow exchange of information among different tools.

For the domain of LCA itself, there are increasing opportunities. Several ongoing projects are focusing on development of Life-Cycle Inventory (LCI) databases, which provide the base for LCA tools (Trusty, 2000; http://www.ivambv.uva.nl/; http://www.uniweimar.de/SCC/PRO/DATA/). These databases are likely to become publicly available in an electronic format over the Internet, and it is expected that their ready availability will increase the use of LCA tools.

In this paper we address the issue of extending an existing building model to address the data needs of an existing tool, using the Athena LCA tool (Trusty and Meil, 1997) as a case study. It is relatively easy to design a *new* LCA tool around appropriate extensions of an existing building model. Extending, however, an existing building model to accomodate the data needs of an *existing* LCA tool presents several challenges. We outline a conceptual model that can address these challenges. Though LCA includes various aspects that are independent of quantification of materials and classification of constructions in a building, in this paper we focus on the materials and construction aspects of LCA.

LCA DATA

One of the fundamental tasks in LCA procedures is the determination of the quantity and type of materials in a building. LCA methods vary but typically involve the use of databases with LCA related data for various materials and/or building components and systems. At the heart of an LCA model is the database, developed and maintained through the LCI (Life Cycle Inventory) process. This process is the critical step that tracks and records the basic resource and waste flows to and from the environment (Trusty, 2000). LCI provides the assembly-specific and site-specific data that an integrated simulation environment would need to include LCA analysis. The site-specificity of the data is easily addressed through basic project inputs such as the city location. The assembly-specificity of the data is more difficult to address, as these data are not resident in the LCI as assemblies, but rather as building materials, manufacturing and construction activities.

The structure of life-cycle information in the Athena model is very specific to particular building assemblies (e.g., wood vs. steel beam and column assemblies) and construction methods. The structure of this information is specific not simply to *categories* of building assemblies, but rather to *individual* assemblies. As an example, Table 1 shows the parameters for the "Wall" object. In an LCA model, each specific wall type (e.g., wood frame, metal frame, concrete blocks, etc.) needs a unique object with a differing set of parameters. Consequently, the LCA object structure is difficult to abstract or fit into common structural frameworks of building data modeling.

The LCA information for each assembly or material also varies with the manufacturers and geographical zones of manufacture, introducing another layer of complexity. For example, a single material such as "recycled aluminium" can be produced by different manufacturers, in different geographical areas, with different methods. The databases need to reflect this and a simulation environment which includes LCA needs to allow selection from multiple databases.

CONCEPTUAL MODEL TO ADDRESS THE GENERAL ISSUES OF INTEGRA-TION

In the previous section we saw how the structure of LCI data is very specific to particular building assemblies and construction methods. On the other hand, shared building models, which form the core of integrated simulation environments, need to be generic to accommodate different views of the same object. The challenge, then, is to incorporate a component-specific structure within a generic building model. In this section we introduce a model or schema, which allows the resolution of this issue. The schema is built upon two levels of abstraction

Table 1: LCA parameters for the "Wall" object.

Object	Parameters
LCA Wood Stud Wall	Assembly name Length Height Openings (area) Stud size Stud spacing Insulation type Sheathing type Finish type
LCA Concrete Block Wall	Assembly name Length Height Openings (area) Block size Rebar size Insulation type Finish type

First level of Abstraction: The Meta Data Schema

At the first level of abstraction is an object-oriented representation of both data and processes, in the form of a meta data schema. The building data are modeled in terms of "building objects" that are related through "relation objects" and are characterized by "parameter objects" (Figure 1). In other words, the attributes of objects and relations between objects are each modeled as software objects. In this way the object model can be expanded in a flexible and modular manner through the creation of new building objects, as well as new relation objects and parameter objects for new and existing building objects. Processes are also modeled as objects, i.e. as "process objects" related to parameter and building objects through input and output relations (Figure 2). A process may vary from a complex simulation engine that accepts a large number of input data and computes a large number of output data, to a simple if-then-else rule with minimal input and output. Data and processes can be added to this environment without restructuring the code because the code operates on a model with a very generic conceptualization (or abstraction) of processes as relations among data rather than on the specific contents of data and processes. This level of abstraction serves as the foundation for the data management and process control provided by this simulation environment (Papamichael et al., 2000).

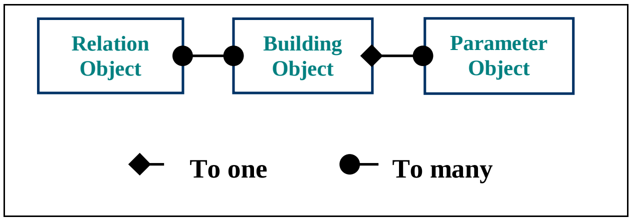

Figure 1: Abstraction of building data as Building objects, Parameter objects, and Relation objects in the Meta Data Schema.

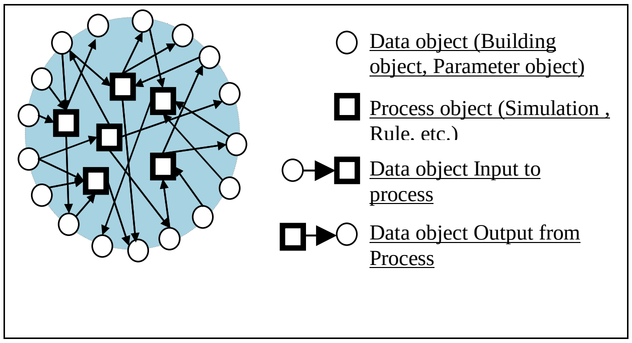

Figure 2. Processes modeled as links among data.

Second level of abstraction: The Building Data Schema

At the second level of abstraction, the Building Data Schema builds on the Meta Data Schema, defining specific instances of building objects such as "spaces", "walls", and "windows". These instances of building objects are related to each other through instances of relation objects such as "composed_of / part_of", "has / owned_by", etc. (Figure 3) Building objects are described by instances of parameter objects such as "u-value", "visible transmittance", "area", etc. Instances of process objects (e.g. simulation tools, rules, data queries) serve as relations among building and parameter object instances. In this way the building data schema addresses the specific needs of the building industry. At any time new instances of building, relation, parameter, and process objects can be defined to address the specific needs of new areas of the building industry. It is conceivable that not just generic building components (e.g. wall, roof) may be defined as instances of building objects but also specific components such as Steel Stud Walls, Open Web Steel Joists and Steel Decking Roof, etc. This becomes necessary when not just the values of parameters associated with these objects are specific to the objects, but the parameters themselves are specific to the objects. This is the case with most LCI data.

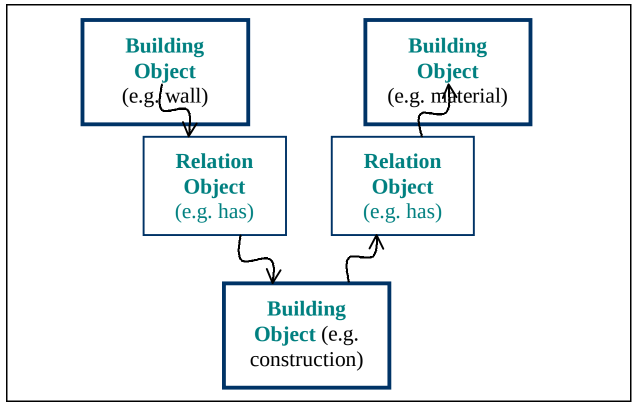

Figure 3: The Building Data Schema is built from lower level abstract objects of the Meta Data Schema.

External Databases of Options for building objects

The definitions of building objects (e.g. space, wall, window) in the building data schema are used to create alternative options for each object, which are stored in external databases. These databases can be distributed and dynamic, that is, available on the Internet and continuously updated by manufacturers of building components and systems, and/or by services and organizations. External databases can be used to select options for building components and systems during the development of the project database for a particular building (Figure 4).

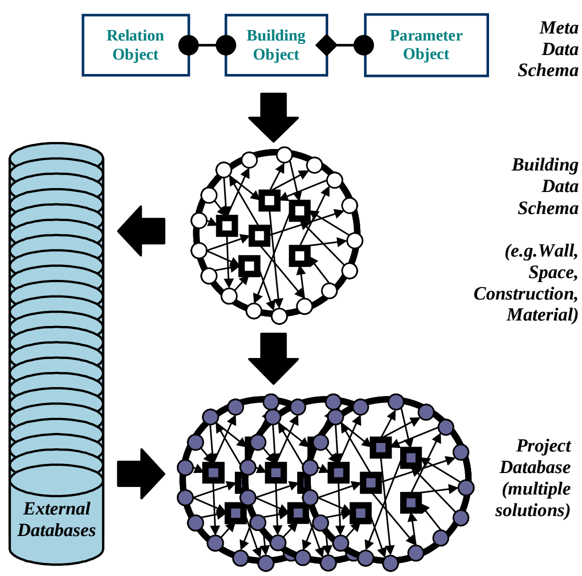

Figure 4. The relationships among the building data schema, the project database, and the external databases.

Because external databases are closely tied to the building data schema, any expansion of the latter must be reflected in the external databases. When parameters are added to the existing building objects in the building data schema, the values for them cannot be automatically available in the external databases of building components and systems. For example, if the schema includes glazing parameters for transmittance and reflectance, and the schema is expanded to include the embodied energy of the glazing, the external databases for glazing will also have to be updated to include values for the new, embodied energy related parameters. When entirely new building objects are added to the building data schema, it will not affect the existing objects in the external databases, though the information for the new objects will, of course, have to be provided in external databases.

LCA MODELING AND DATA MANAGEMENT

During the course of a single project, run-time objects, relations and parameters can be created and deleted as needed, managed by a process control mechanism. The state of the objects and relations describing the project can be stored (i.e. made to persist) in a project database, which can hold multiple alternative states, for the whole building project. External databases, on the other hand, store multiple possible states (or options) for each object. The issues involved with these two aspects of simulation environments, i.e., modeling of a specific building on the one hand and the external databases of building object options on the other, are different though inter-related. This section describes how the object model described in the previous section resolves these issues for materials LCA.

Data Schema Extensions to Support Materials LCA

In the previous section we described how the use of a data meta-schema with the abstraction of all building data and related processes into inter-related software objects allows the expansion of the building model through the definition of new objects as data and processes in a database. What this means for LCA modeling is that each individual building assembly can be added as a new building object to the building data schema, without breaking the existing schema. The extension does not require redefinition of existing building objects such as walls, roofs, floors, etc. Instead, specific wall objects (Steel Stud Walls) and roof objects (Open Web Steel Joists and Steel Decking Roof) are defined in addition to generic wall and roof objects.

When a user creates an object, e.g, a wall, a wall construction prototype is selected by default or by the user from the options available in the external database of wall options. The building data schema is queried for the list of parameters required for the wall construction object. Run-time parameters are accordingly created and placed under the construction object. Following this, the external databases are queried and run-time values are created and loaded for each parameter. More generally, the process control mechanism operates upon abstract building objects, relations and parameters rather than upon specific building objects (e.g. walls, roofs), relations (e.g. part_of / composed_of), or parameters (e.g. wall u-value). Therefore, the addition of new objects, relations and parameters does not break the process control mechanism. The code for the process of loading, creating, destroying, or saving objects and relations does not require restructuring upon expansion of the building data schema, so the building model can be expanded incrementally as needed through addition of new data objects and processes. This expansion capability is very useful when a large number of specific objects need to be added to the building data schema, as in the case of modeling LCA data.

LCA Database Management

In the data schema described above, there is provision for flexibility of the structure of the building model. For example, depending upon the construction prototype selected, the parameters associated with the construction object may be different. Since this flexibility is made possible by the data schema, we need not worry about including all LCA related construction data into a single generic construction object. If construction type A needs "thickness", "rebar size" and "concrete density" as parameters, and construction type B needs "stud size", "stud spacing" and "sheathing type" as parameters, we need not combine all these parameters into a generic construction type. We can allow them to be two separate construction objects. The concept of separating one physical object with differing semantic representations into different software objects (usually referenced through a tag or a label) has been usefully applied in other integrated environments (Mahdavi et. al, 1996). This is helpful in the maintenance of the external databases of options (or prototypes). When new objects are added to the database, values need to be provided only for the options for the new object. The existing objects remain largely untouched. Thus, the database is expandable, and maintenance is relatively easy and straightforward. By contrast, if we had added LCA parameters to existing construction objects, the entire existing database of construction prototypes/options would have to be modified.

The additional level of abstraction, i.e., the Meta Data Schema, simplifies the separation of different construction types into different construction objects by making the addition and management of these objects particularly easy, clean and streamlined. Even though the objects are different, their management by the code isn't. The code is independent of the actual contents of the object. It operates upon objects and parameters rather than upon "walls" and "stud spacing".

<u>An example of implementation</u>

An initial version of the data schemas (i.e. the meta data schema and the building data schema) has been implemented in the Building Design Advisor (BDA) software (Papamichael et al., 1997). The BDA already has links to the DOE-2.1E energy simulation tool (Birdsall et al., 1990), the DELight daylighting simulation tool (Hitchcock, 1995), and ECM, a new, simplified electric lighting calculation tool. Links with Radiance, an advanced lighting simulation tool (Ward and Shakespeare, 1998), are almost complete. Recently, attempts were made to identify the requirements and means for integrating Athena, a materials LCA tool (Trusty and Meil, 1997). Athena has approximately 25 "Assembly Types" (e.g. Wide Flange Beams and Hollow Steel Section Columns, Open Web Steel Joist and Steel Decking Roof, Concrete Footing, Concrete Cast-in-place Wall, etc.), which are categorized into 5 "Assembly Groups" (Beams and Columns, Extra Basic Materials, Floors and Roofs, Foundations, Walls). In BDA, walls, floors and roofs are represented by the generic "boundary" object. The boundary "contains" one or more "boundary_segment"s. The boundary segment "has" a "construction". Table 2 identifies the objects that exist in BDA for the space boundaries, and their existing parameters. Table 3 shows a possible scenario for the expansion of the BDA model to accommodate the Athena Assembly Types relevant to floors, roofs, and walls.

In the above scenario, the only LCA specific parameter to be added to the existing construction object is the "LCA Assembly Name". Then, depending upon the construction prototype selected by the user or defaulted by the software, an object in the building data schema corresponding to that name is created.

Currently, there are no objects for foundations, beams and columns in the BDA building data schema. These can be added following a principle similar to that applied for walls, roofs and floors. A possible scenario for foundations is shown in Table 4.

<u>Multi-level output</u>

LCA output such as graphs and tables for embodied energy, resource use, emissions to air, emissions to water, global warming potential, etc. can be generated at different levels of resolution. They can be computed for a particular building component such as a single wall or roof, for a particular space such as a large open-plan office space, for a single story, or for an entire building or complex. The data schema outlined in this paper can easily accommodate this multi-level output. The various LCA output parameters can be added to the building objects at various levels, i.e., at the building level, floor level, space level, and component level. The values for these parameters do not need to be present in the external databases. Instead, they will be computed by the process (in this case Athena), which has "output" links/relations with them.

Table 2: Existing BDA Objects and Parameters That Describe the Surface Boundaries Such As Walls, Floors, and Roofs

Object	Relation	Parameters
Boundary	*contains*	Length Height or Width
one or more **Boundary segment** *which*	*has*	Surface Area
one **Construction**		Inside film resistance Effective thermal resistance Effective thermal conductance Boundary type

Table 3: Expansion of the BDA Object Model to Accommodate Athena Wall, Floor, and Roof Assembly Types

Object	Relation	Parameters
Boundary	*contains*	Length Height or Width
one or more **Boundary segment** *which*	*has*	Surface Area
one **Construction** *which*	*has*	LCA Assembly Name Inside film resistance Effective thermal resistance Effective thermal conductance Boundary type
zero or one Concrete, cast in place		Thickness Rebar Size Concrete Density
zero or one Wood studs		Stud size Stud spacing Sheathing Type
zero or one OWSJ & Steel decking (Roof)		Span distance Live load
zero or one Steel Joists with Wood Decking (Flooring System)		Decking type Decking thickness Steel gauge Joist type
…		

Table 4: Expansion of the BDA object model to accommodate Athena foundation types

Object	**Relation**	**Parameters**
Foundation *which*	*has*	Length Width Thickness
one Foundation construction *which*	*has*	LCA Assembly Name
zero or one Concrete footing		Rebar Size Concrete Density
zero or one Concrete slab on grade		Concrete Density

UNRESOLVED ISSUES

The above discussion has touched upon some of the complexities of modeling and simulation in the building industry, specifically in the context of materials LCA. However, there are several unresolved issues relating to the multiple functions of particular building elements, the multiple ways in which elements and their functions may be aggregated, the multiple assumptions about their form and function to simplify calculations, and the multiple sources of information about these elements.

Each building component and system can be viewed from a number of perspectives, depending on the domain of interest. For example, a wall can be a thermal barrier with a u-value or a structural element with a load-bearing capacity, or it can represent resource use and emissions to water and air in its construction and demolition. If there is significant overlap between several views, it might make more sense to have a representation of the component which is a union of these views. In other cases it might make more sense to have the same physical object represented as different software objects, each with its own semantic properties. However, there are no hard and fast rules about which approach is better in any individual case.

Sub-classing and inheritance may also be used to address some of these issues. However, different situations may require different heirarchical structures for inheritance. This is especially true when building components and their semantic properties may be aggregated to form other building components in multiple ways. For example, a glass curtain wall may be modeled as one integrated component with an overall u-value for simplified thermal calculations, or it may be separated into glazing and metal structure for structural calculations, while each mullion and glazing piece may be modeled as separate entities for daylighting calculations. Beams and columns will need to be modeled as separate entities for structural calculations, but LCI data might only be available for the beam and column assembly as a whole because the processes involved in the construction of these assemblies is specific to beam and column type combinations.

Within each domain of interest, a component may be modeled at different levels of detail. For example, an HVAC system could be modeled at the component level, with calculation of pressure drops in ducts and energy transfer between finite elements or nodes, or it could be modeled as an overall efficiency value. Sometimes, simplifying assumptions, rather than reducing the amount of information necessary for computation, require more information in the form of look-up tables and curves. An integrated simulation environment which allows for modeling at both levels, possibly with different tools, needs to allow for a building representation that is compatible with multiple levels of detail.

Several of these issues come together and surface rather prominently when attempts are made to standardize the formats for the component databases provided by manufacturers, service providers, and national or international A/E/C organisations. This picture is further complicated by the fact that some required information, such as the specifications and performance of particular components, is considered to be proprietary and confidential. Some component properties require considerable infrastructure to measure and may not be provided by all manufacturers of that component. The IAI (International Alliance for Interoperability) was formed specifically for the purpose of sharing data among organisations and tools, and the development of the IFCs (Industry Foundation Classes) is well underway. The building industry is moving towards better interoperability and integration, and some of these issues are being resolved to give us a road map for guidance, and some others will need to be resolved as they are encountered along the way.

CONCLUSIONS

In this paper we have described a conceptual model for an integrated simulation environment based on two levels of abstraction which allows for incremental expansion. Both data and processes may be added to this environment without the need for restructuring the existing environment. We described how this is especially helpful in incorporating LCA tools within the integrated environment, since materials LCA requires the definition of several specific construction objects. We demonstrated how the concept will work with an example, using an environment (Building Design Advisor) developed with this concept, and a LCA tool Athena.

Significant work is needed to address several larger issues of integration and we hope that the thoughts and modeling approaches presented in this paper will contribute towards addressing these larger issues.

ACKNOWLEDGEMENTS

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology, State and Community Programs, Office of Building Research and Standards of the U.S. Department of Energy.

REFERENCES

Birdsall, B.E., Buhl, W.F., Ellington, K.L., Erdem, A.E. and Winkelmann F.C., "Overview of the DOE-2 building energy analysis program, version 2.1D." Lawrence Berkeley Laboratory report LBL-19735, Rev. 1, Berkeley, CA., 1990.

Hitchcock, R. J., "Advancing lighting and daylighting simulation: the transition from analysis to design aid tools", Proceedings of Building Simulation '95, International Building Performance Simulation Association, 1995.

Jokela, M., Keinänen, A., Lahtela, H., and Lassila K., "Integrated building simulation tool RIUSKA," Proceedings of Building Simulation, Prague, Czech Republic, 1997.

Klein, S. A., Duffie, J.A., and Beckman, W.A., "TRNSYS - A Transient Simulation Program," ASHRAE Trans, 82, 623, 1976.

Mahdavi, A., Mathew, P., Lee, S., Brahme, R., and Kumar, S., "On the structure and elements of

SEMPER," Proceedings of ACADIA ’96 Conference, Tucson, Arizona, October 31 – November 2, 1996.

Papamichael, K., Pal, V., Bourassa, N., Loffeld, J., and Capeluto, G. "An Expandable Software Model for Collaborative Decision Making during the Whole Building Life Cycle", Proceedings of ACADIA 2000 Conference, Washington D.C., October 19-22, 2000.

Papamichael, K., LaPorta, J., and Chauvet, H., "Building Design Advisor: automated integration of multiple simulation tools." Automation in Construction, Vol. 6, pp. 341-352, 1997.

Pohl, J., LaPorta, J., Pohl K.J., and Snyder J., "AEDOT Prototype (1.1): An Implementation of the ICADS Model." Technical Report CADRU-07-92, CAD Research Unit, Design Institute, School of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA., 1992.

Trusty, W. B., "Let's Talk Data", Athena SMI Newsletter, Vol. 1, No. 3, December 2000.

Trusty, W. B. and Meil, J. K., "ATHENA™: An LCA Decision Support Tool - Application, Results and Issues" - Proceedings: Second International Conference on Buildings and the Environment. Sponsored by CSTB and CIB TG8, Paris, France, June 1997.

Ward, G. and Shakespeare, R., "Rendering with Radiance: The Art and Science of Lighting Visualization", Morgan Kaufman, 1998.